The Scientists and the Bar in Santa Fe: A New FABLE(S)

Richárd Legéndi[1], Rajmund Bocsi[1], László Gulyás[1,2]

[1] AITIA International Inc, Czetz János utca 48-50, 1039 Budapest, Hungary
[2] Loránd Eötvös University, Pázmány Péter sétány 1/c, 1117 Budapest, Hungary

{rlegendi, rbocsi, lgulyas}@aitia.ai

ABSTRACT

The Functional Agent-Based Language for Simulations (FABLES) is a special purpose language for agent-based modeling that is intended to reduce programming skills required to create simulations. The aim of FABLES is to allow modelers to focus on modeling, and not on programming.

The original idea of FABLES was presented at SwarmFest 2003 in Notre Dame, listing our goals to create a simulation language that

- requires minimal programming skills,
- is able to describe the model focusing on the nature of the model, and leaves the implementation to the compiler,
- is general enough to accommodate possibly any agent-based model, but focuses on the common techniques and methods,
- supports the creation, observation and control of agent-based simulations with the help of interactive wizards,
- has a formalism similar to the mathematical formalism used in publications related to agent-based modeling. In these papers there's no space to publish algorithms and thus models are described by formulas and quantors.

At SwarmFest 2005 in Turin, Italy a proof-of-concept version was shown, while at the Agent 2007 conference in Chicago, we introduced the first stable release of the FABLES compiler and its integrated environment that merges the attributes of object-oriented, functional and procedural languages to provide flexibility in model design. In particular,

- The properties of the model, i.e., parameters, constants, global relations (facts), etc. are specified in a declarative way. This part of FABLES follows a functional paradigm that is similar to the formulas of used in scientific papers.
- The modeler specifies the basic structure of the agents used in the simulation, including their properties (variables, behavioral functions, etc.). Agents are considered as independent entities, thus they are described using the object-oriented paradigm.
- Agent interactions are, again, defined using functional formulas.
- State transitions (i.e., changes in variables) are defined by assignments, i.e., in an imperative way.
- The dynamics of the model (i.e., the activation of state transitions and interactions) is specified with the use of schedules (just like in Swarm, Repast or MASON). These powerful constructs of the discrete time, discrete event modeling can also be viewed as an application of the imperative programming paradigm.

This paper works with an improved, new release of 2009 and demonstrates its capabilities and concepts by presenting an implementation of the well known El Farol Bar model.

ACKNOWLEDGMENTS

The work reported here benefited from a grant from the Hungarian Government (grant #GVOP-3.2.2-2004.07-005/3.0). The partial support of the European Commission via the FP6 STREP projects QosCosGrid (contract #033883) and EMIL (contract #033841) is also gratefully acknowledged. László Gulyás also acknowledges the support of the Hungarian Government (OTKA T62455).